

# Microsoft DevOps Solutions: Implementing Orchestration Automation Solutions

---

PLAN AND AUTOMATE RELEASE PROCESS



**Daniel Krzyczkowski**

MICROSOFT MVP & SOFTWARE DEVELOPER

@DKrzyczkowski [www.techmindfactory.com](http://www.techmindfactory.com)



# Course Overview



Combine release targets depending on release deliverables

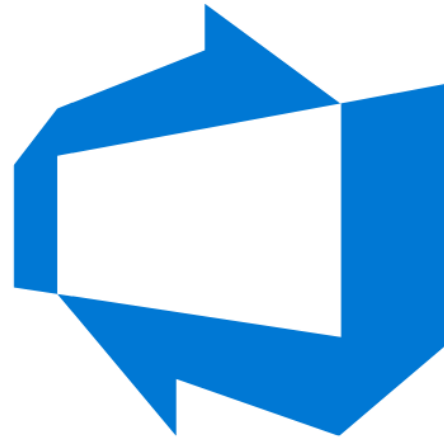
Design the release pipeline to ensure reliable order of dependency deployments

Organize shared release configurations and process with YAML files and variable groups

Design and implement release gates and approval processes

Summary





Azure DevOps



# Continuous Delivery and Deployment

---



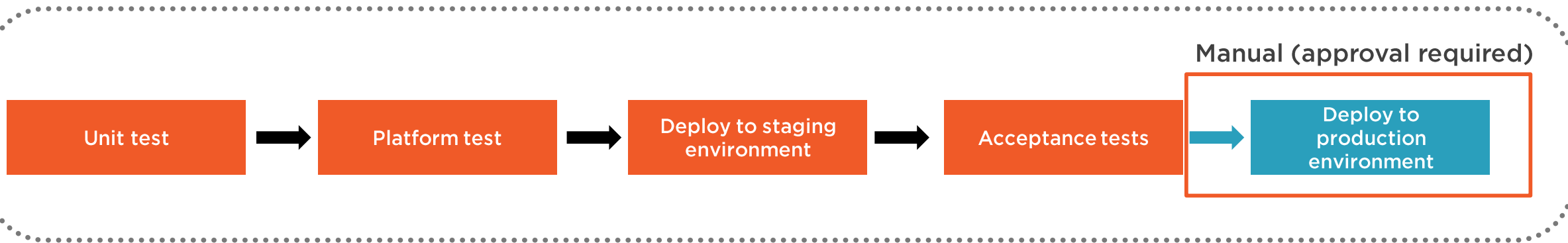
# Continuous Delivery

A set of processes, tools, and techniques that enable fast, reliable, and continuous delivery of software.

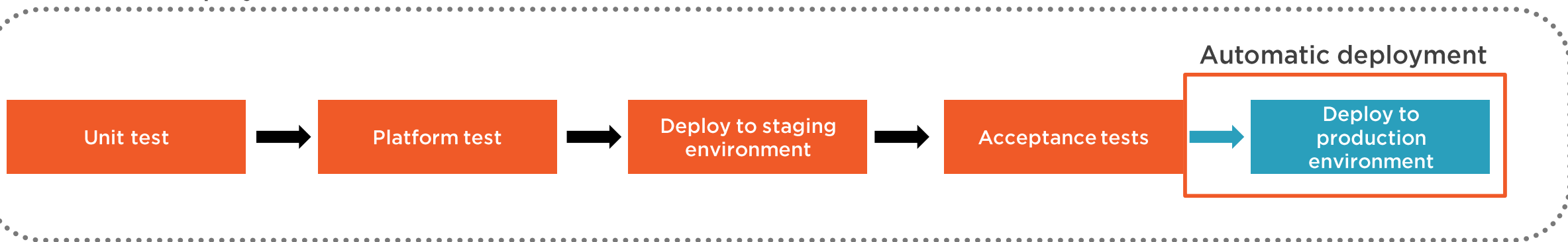


# Continuous Delivery vs. Continuous Deployment

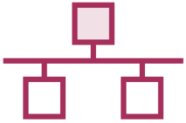
## Continuous Delivery



## Continuous Deployment



# Release Deliverables



Infrastructure code – Azure Resource Manager (ARM) templates



Web, desktop, or mobile application packages



Images and files



Database migration scripts



# Release Targets



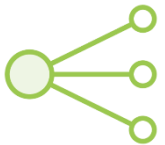
A managed service, such as Azure App Service



A serverless environment, such as Azure Functions



A physical machine or virtual machine



A containerized environment, such as Azure Kubernetes Service





# Different Kinds of Environments

## DEV

Environment with all artifacts deployed. Developers can work and test using this environment

## TEST

Environment used by QA team to test all the features delivered by development team

## PRODUCTION

Environment that is available for end users with all deployed solution's components



Release pipeline may  
consist of infrastructure and  
applications package's  
deployment steps



# Design the Release Pipeline

---



# DevOps Automation

Build Web App



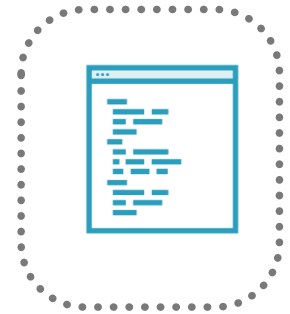
Build Web API



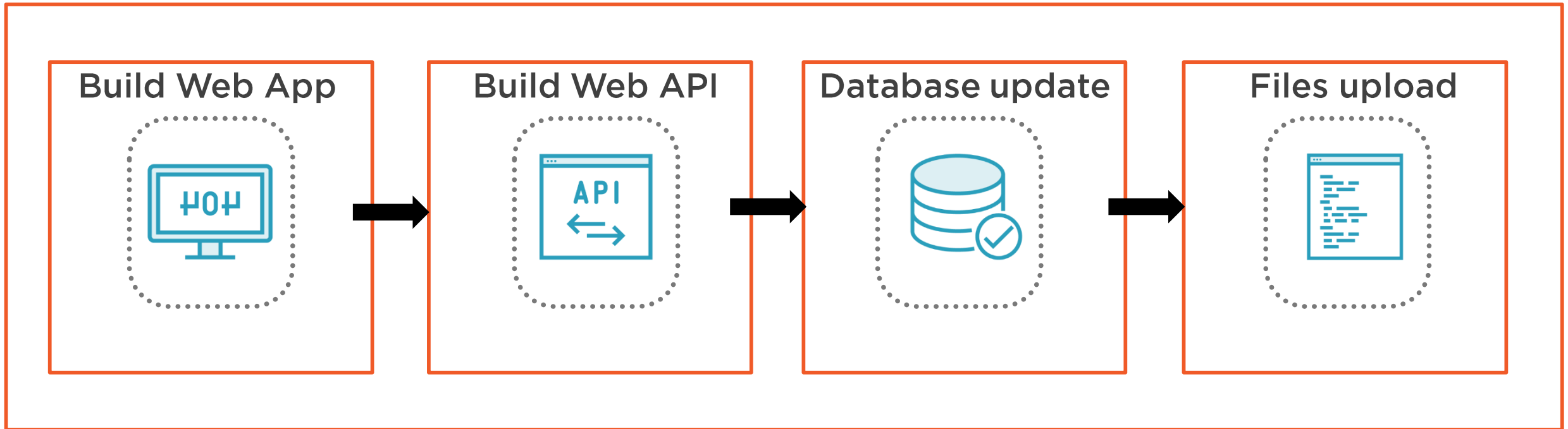
Database update



Files upload



# DevOps Orchestration

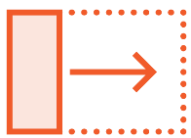


# DevOps Orchestration

Automation applies to tasks that are common to one area, such as launching a web service, or integrating a web app, or updating a database. When all these tasks must work together, DevOps orchestration is required.



# Parts of a Basic Continuous Delivery Pipeline



**Trigger** – an action that triggers continuous delivery pipeline.  
Example: new commit to specific branch in the source code repository.

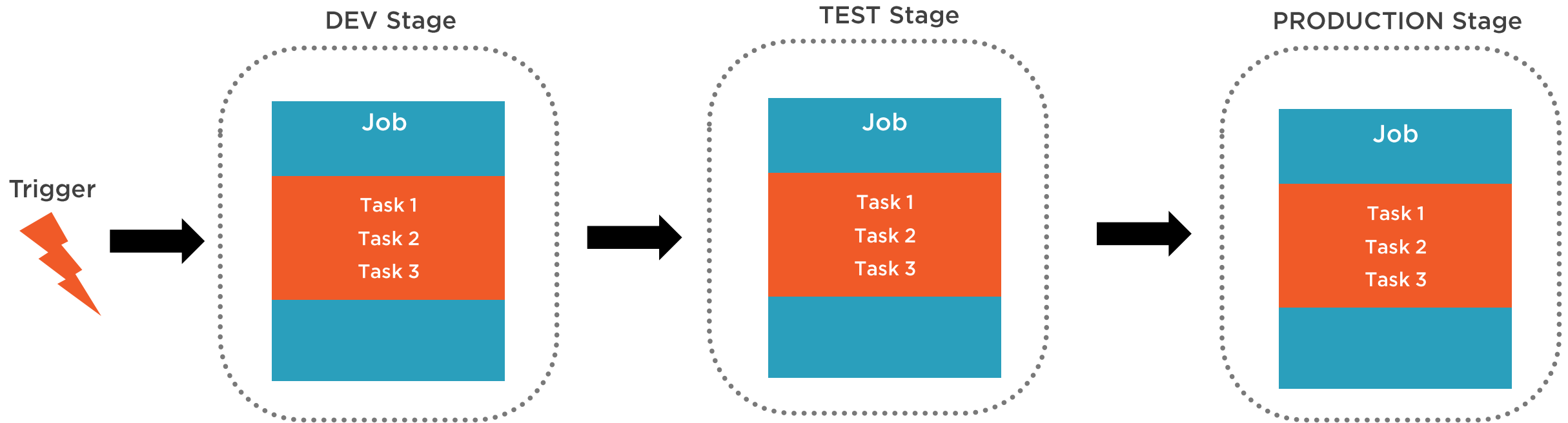


**Deployment phase (stage)** - a logical boundaries in the pipeline. A stage is made up of jobs.  
Example: stage that deploys the application packages to a specific environment (dev or production).



**Job** - a series of steps (tasks) that defines how to build, test, or deploy software.

# Release Pipeline Example





# Demo

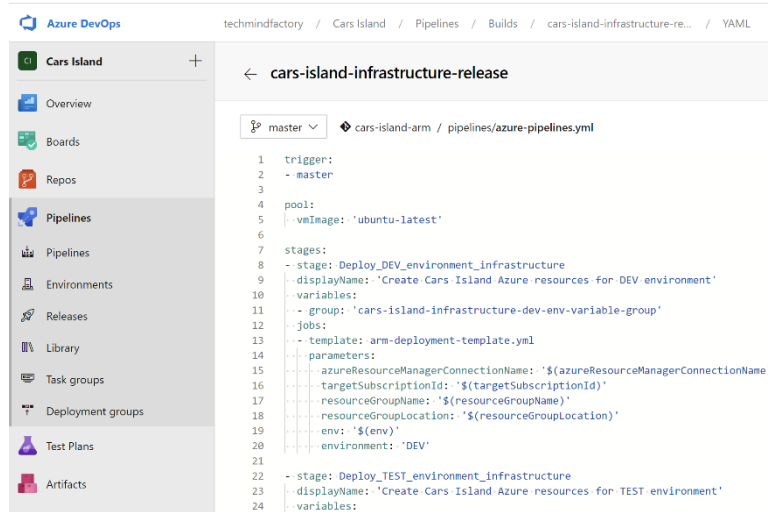


## Implement infrastructure release pipeline using Azure DevOps Releases

- Create release pipeline for the Azure infrastructure
- Setup variable groups to keep shared configuration



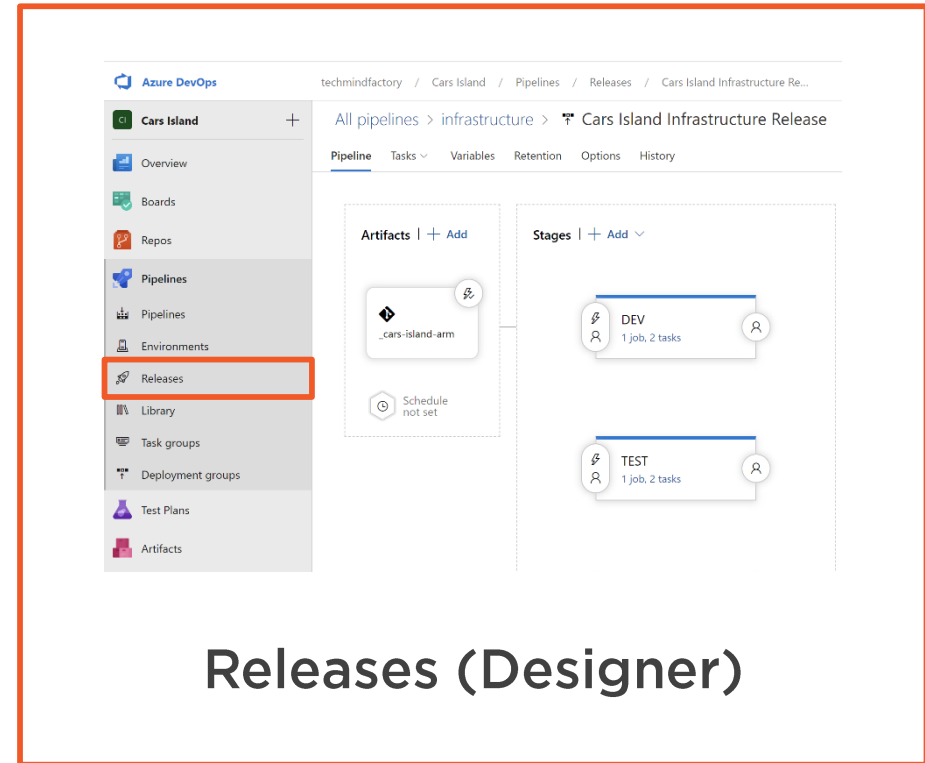
# Azure DevOps Pipelines and Releases



The screenshot shows the Azure DevOps interface for editing a pipeline. The left sidebar contains navigation links: Overview, Boards, Repos, Pipelines (selected), Environments, Releases, Library, Task groups, Deployment groups, Test Plans, and Artifacts. The main area displays the 'cars-island-infrastructure-release' pipeline in YAML format. The YAML content is as follows:

```
1 trigger:
2   - master
3
4 pool:
5   - vmImage: 'ubuntu-latest'
6
7 stages:
8   - stage: Deploy_DEV_environment_infrastructure
9     displayName: 'Create Cars Island Azure resources for DEV environment'
10    variables:
11      - group: 'cars-island-infrastructure-dev-env-variable-group'
12    jobs:
13      - template: arm-deployment-template.yml
14        parameters:
15          azureResourceManagerConnectionName: '${azureResourceManagerConnectionName}'
16          targetSubscriptionId: '${targetSubscriptionId}'
17          resourceGroupName: '${resourceGroupName}'
18          resourceGroupLocation: '${resourceGroupLocation}'
19          env: '$(env)'
20          environment: 'DEV'
21
22   - stage: Deploy_TEST_environment_infrastructure
23     displayName: 'Create Cars Island Azure resources for TEST environment'
24     variables:
```

Pipelines (YAML files)

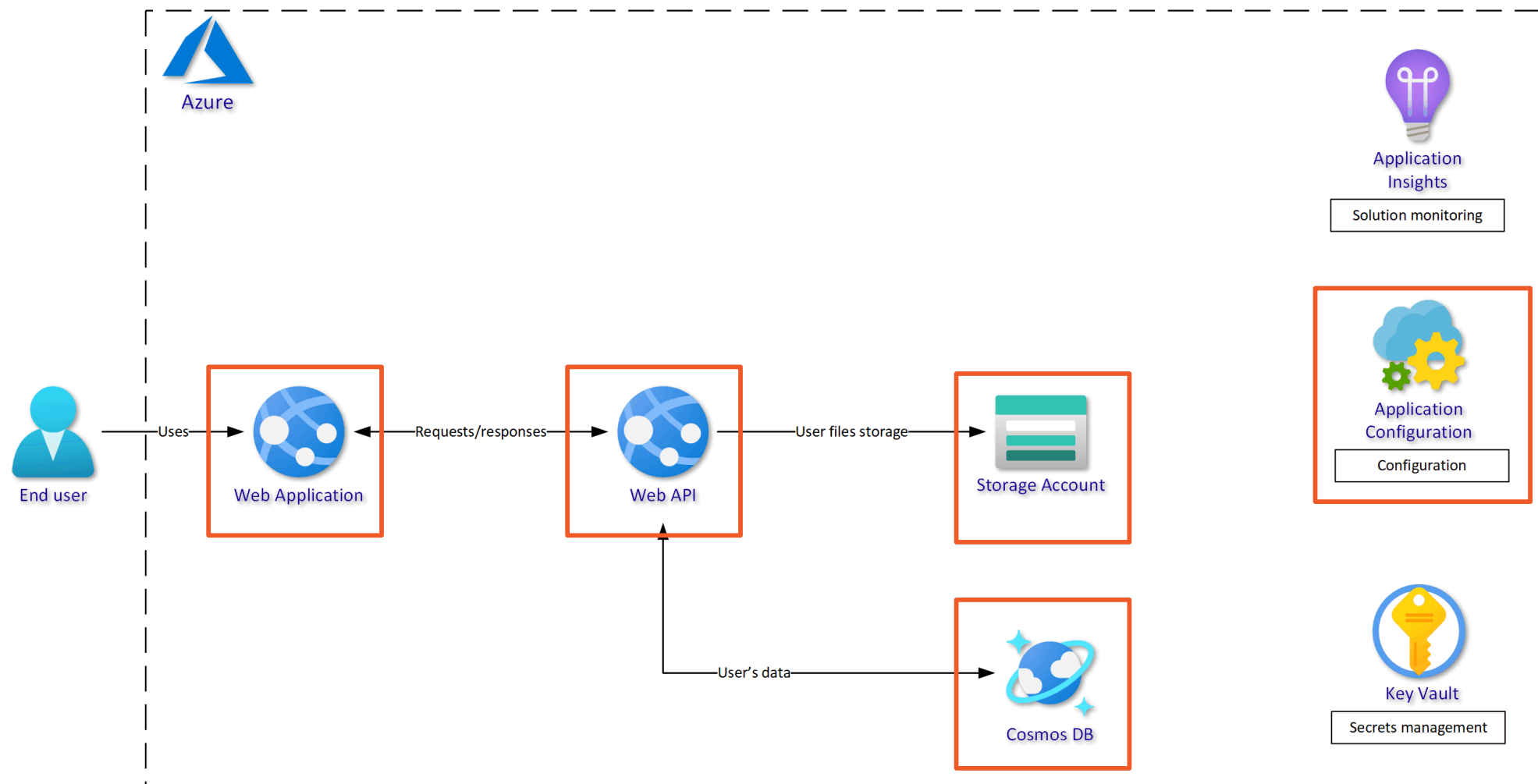


The screenshot shows the Azure DevOps Releases Designer interface. The left sidebar is identical to the Pipelines view. The main area displays the 'Cars Island Infrastructure Release' release configuration. The 'Artifacts' section shows a dependency on the '\_cars-island-arm' artifact. The 'Stages' section shows two stages: 'DEV' (1 job, 2 tasks) and 'TEST' (1 job, 2 tasks). The 'Releases' link in the left sidebar is highlighted with a red box.

Releases (Designer)



# Solution Architecture



# Demo



## Implement web apps release pipeline using Azure DevOps Releases

- Create release pipeline for the ASP .NET Core web app
- Update settings with configuration from the Azure App Configuration



# Summary



Difference between Continuous Delivery and Deployment

DevOps automation vs DevOps orchestration

How to combine release targets depending on release deliverable

How to design the release pipeline to ensure reliable order of dependency deployments

How to organize shared release configurations using variable groups

